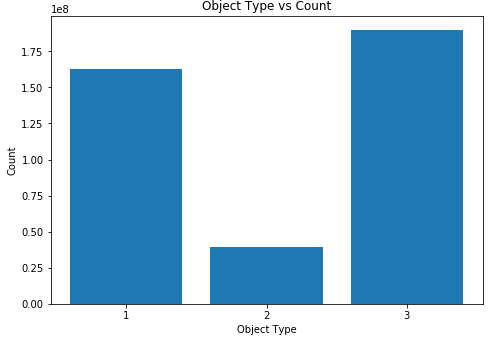
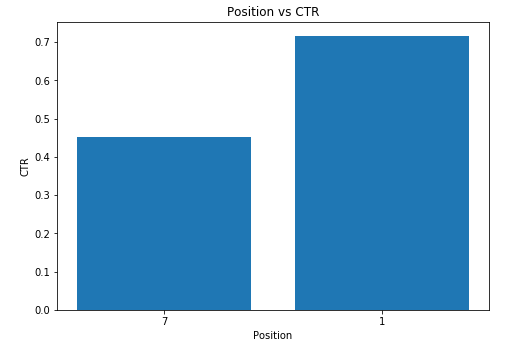
EDA:

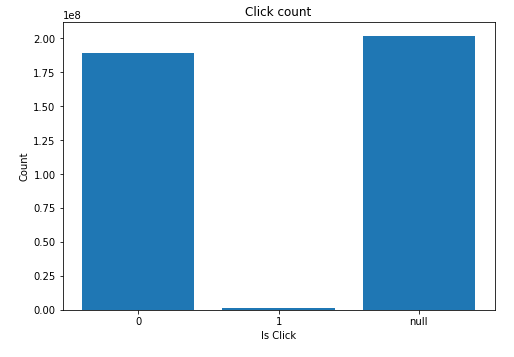
* Total entries in the train file – 392356948
* Count of each Object type below. We can observe 1 and 3 being predominantly higher compared to 2.



* Position vs CTR for contextual ads. Position 1 tends to have a higher CTR compared to Position 7. We can target Position 1 more often to increase our CTR.

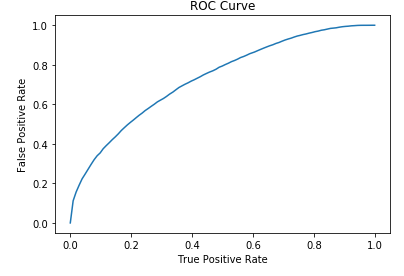


* Distribution of Clicks data. The data is imbalanced with majority of non-click data.



Feature Engineering and Modeling:

* There are lot of variables to perform EDA like User browsing details, search details, ad info, etc. But since the data size is huge (~20GB) it was taking lot of time to run each query on local system as the resources are limited. Joining all the tables itself took one day. I thought joining full data was essential as taking few samples from each file and then joining would result in very few data points and no data points.
* So for modeling and feature engineering purposes I have sampled the data after joining all the tables and provided the approach for basic LR modeling on PySpark.
* LocationID is null in the train data for all the records. So Considered Location info from Search data after joing with the train data.
* Size of the sampled data – 1844227
* Feature details
  + Categorical features considered - *'Position','ParentCategoryID','SubcategoryID', 'IsUserLoggedOn','UserAgentID', 'UserAgentOSID','UserDeviceID','UserAgentFamilyID', 'RegionID', 'CityID'*
  + Numeric features - *'HistCTR', 'Price'*
  + Label - ‘IsClick’
* Converted all the categorical features into one-hot encoded features and trained the LR model on top of all the features with *IsClick*  as label.
* Performance



* + *Training set areaUnderROC: 0.73245018752254*
  + *Test Area Under ROC 0.6136698146103157*
* Next Steps:
  + We can run more steps of EDA on multiple features on a distributed platform to provide more insights to the website so that they can increase their conversions.
  + As part of this POC, few important features were left out like searchParams, Ad Title, Ad Params, etc. We can engineer those features and run the model on entire data rather than sample of data on a distributed platform.
  + We can apply regularization techniques to reduce the number of features and the model complexity (there by overfitting).
  + Since the data is imbalanced, we can try different sampling techniques (undersampling, oversampling, SMOTE) before modeling the data.
  + Once the model is fully trained we can check the effect of each feature and make a list of recommendations to website owner to improve the performance of their website.